AF IFW

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 15

Application Number	09/776,040
Filing Date	FEBRUARY 1, 2001
First Named Inventor	JOERG EHRHARDT
Art Unit	2174
Examiner Name	Peng Ke
Attorney Docket Number	7057 US

ENCLOSURES *(Check all that apply)*

- [x] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below): Return Post Card

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	TEKTRONIX, INC.		
Signature	Francis I. Gray		
Printed name	FRANCIS I. GRAY		
Date	NOVEMBER 29, 2005	Reg. No.	27,788

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Pauline L. Bradley		
Typed or printed name	Pauline L. Bradley	Date	November 29, 2005

PTO/SB/17 (12-04v2)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($)	500.00

Complete if Known	
Application Number	09/776,040
Filing Date	FEBRUARY 1, 2001
First Named Inventor	JOERG EHRHARDT
Examiner Name	Peng Ke
Art Unit	2174
Attorney Docket No.	7057 US

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account Deposit Account Number: 20-0352 Deposit Account Name: TEKTRONIX, INC.

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☑ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims		Extra Claims	Fee ($)		Fee Paid ($)
_____	- 20 or HP =	_____	x _____	=	_____

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims		Extra Claims	Fee ($)		Fee Paid ($)
_____	- 3 or HP =	_____	x _____	=	_____

HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets		Extra Sheets		Number of each additional 50 or fraction thereof	Fee ($)		Fee Paid ($)
_____	- 100 =	_____	/ 50 =	_____ (round up to a whole number)	x _____	=	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other (e.g., late filing surcharge): APPEAL BRIEF	$ 500.00

SUBMITTED BY

Signature	Francis I. Gray	Registration No. (Attorney/Agent) 27,788	Telephone 503 627-7261
Name (Print/Type)	FRANCIS I. GRAY		Date NOVEMBER 29, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **JOERG EHRHARDT, JENS KITTAN and WOLFGANG BORGERT**

Art Unit: **2174**

Serial No.: **09/776,040**

Examiner: **Peng Ke**

Filed: **February 1, 2001**

For: **SETTING UP A COMMUNICATION PROCEDURE BETWEEN INSTANCES AND A PROTOCOL TESTER USING THE METHOD**

November 29, 2005

Mail Stop Appeal Brief - Patents
COMMISSIONER OF PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

APPEAL BRIEF

Dear Sir:

This is an appeal from the Examiner's rejection dated June 28, 2005 in the above-identified application finally rejecting claims 1-13 over prior art.

Real Party in Interest

The real party in interest in this appeal is Appellants' assignee, Tektronix International Sales GmbH of Schaffhausen, Switzerland.

Related Appeals and Interferences

There are no prior or pending appeals, interferences or judicial proceedings known to Appellants or Appellants' legal representative or assignee which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-13, the only claims in this application, all stand finally rejected and are the claims being appealed.

Status of Amendments

There were no amendments to the claims filed subsequent to the final rejection in this application.

Summary of Claimed Subject Matter

The presently claimed invention relates to setting up a communication procedure between instances of a communication network – more particularly between a protocol tester as one instance and an item under test of the communication network as another

instance. Specifically the protocol tester emulates a protocol layer for testing a specified protocol layer of the item under test based on the communication procedure. Abstract communication interfaces of the emulated protocol layer are selected and communication data contained in description files is selected for exchange at the abstract communication interfaces. The communication procedure is set up based upon the emulated protocol layer, the abstract communication interfaces and the communication data, with parameters for the abstract communication interfaces and the communication data being determined graphically. (Page 2, line 19 - page 3, line 5) As is well known by those skilled in the art, protocol layers refer to the various levels of an OSI protocol stack – generally there are seven layers from the physical layer to the application layers that make up the protocol stack. In order to test a specific protocol layer of the protocol stack, data has to be communicated from an adjacent protocol layer via appropriate abstract communication interfaces to the specific protocol layer.

Fig. 1 shows a graphical user interface (GUI) **10** that allows graphically selecting instances to take part in the communication protocol, one of the instances being a protocol tester emulating a component **TC_1**. (Page 4, lines 2-18) Fig. 2 shows the GUI for selecting the protocol layer to be emulated by the protocol tester – "isdn12". (Page 4, line 19 - page 5, line 4) Fig. 3 shows the GUI for selecting the service access point (SAP) or abstract communication interface – "TS1low". (Page 5, lines 5-8) Fig. 4 shows the GUI for selecting the communication data from message pools. (Page 5, lines 9-12) Fig. 5 shows the GUI that provides a user with various types of information as a summary of the steps taken via Figs. 1-4, as well as a graphical representation of the resulting portion of the communication procedure. (Page 5, lines 13-16; page 6,

lines 1-5) Fig. 6 shows the GUI illustrating how the user may graphically set up the communication procedure, including the possibility of incorporating codes in a specified programming language (Forth) into a block by using an entry mask. (Page 5, lines 17-24) Finally Fig. 7 shows the GUI for an isdn-PDU setup for setting up a message from the tester to the instance under test. (Page 6, lines 6-10) Annex A1 shows the code automatically generated by the tester. (Page 6, lines 22-23).

Grounds of Rejection to be Reviewed on Appeal

(1) Whether claims 1-3, 6-10 and 13 are anticipated by Swift et al ("Swift") under 35 U.S.C. 102(b).

Argument

(1) 35 U.S.C. 102(b)

35 U.S.C. 102(b) provides in pertinent part that a "person shall be entitled to a patent unless – . . . the invention was patented or described in a printed publication in this or a foreign country . . . more than one year prior to the date of the application for patent in the United States." It is axiomatic that the reference, to be anticipatory, has to disclose all of the elements recited in the rejected claim in the same relationship.

Discussion of Swift Reference

Swift discloses a multi-protocol message sequence generator that enables a user to define a sequence of messages and transmit the messages to a target network object for testing, which target object is a network management system that is responsive to the reception of the message sequence. The message sequences correspond to actual message sequences transmitted by network source objects to target objects in a production network. Network sources correspond to switches, routers, bridges, repeaters, etc. – any device capable of communicating messages on a network. A graphical user interface (GUI) is provided that allows the user to simply select the message type, content and sequencing the user wishes to generate. The user may also edit existing messages. As a result the user has a wide variety of options for creating testing scenarios in a quick, easy and efficient manner.

As shown in Fig. 1, a sequence generator **102** is connected to a communications network **103** to which also is connected a data collector **108**. One or more store forward files **110** are accessible by the data collector. Each store forward file is monitored by a data distributor **112** which passes on a message sequence **106** to a target object **114**. Fig. 2 shows a message sequence generator **102** that includes a message sequence engine **218** that communicates with a GUI **212**, a message database **214** (containing actual messages previously sent) and a message text file **216** (similar to message database, but creatable using standard word processing) and provides the message sequence via a network interface **224** to the communications network, preferably a TCP/IP network, i.e., a network that uses the transmission control protocol for

robustness of data transmission and the internet protocol for transmitting data from location to location or node to node. The message sequence engine allows the user to create a message sequence definition **222** via the GUI. The message sequence definition defines the message sequence.

As with all other communications protocol, TCP/IP is composed of layers:

- **IP** - is responsible for moving packets of data from node to node. IP forwards each packet based on a four byte destination address (the IP number). The Internet authorities assign ranges of numbers to different organizations. The organizations assign groups of their numbers to departments. IP operates on gateway machines that move data from department to organization to region and then around the world.

- **TCP** - is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.

It is noted that Swift does not discuss emulation of a protocol layer and, aside from the reference to the TCP/IP network as the transmission medium, does not mention any protocol layers at all. The TCP/IP network is merely the medium for transferring the message sequences from the network sources (message generator) to the target objects. The purpose of the Swift invention is to test target objects, not to test any specified protocol layer.

Argument

Claims 1 and 8:

The Examiner argues that Swift teaches a method of setting up a communication procedure between instances that include selecting the instances that take part in the communication procedure, one instance being a protocol tester and another instance being an item under test, citing page 1, paragraph 3, lines 1-9; page 6 paragraph 3; and page 8, paragraph 5 – A network management system receives "events (messages) from a wide variety of network components, such as network switches and network routers" to which the management system responds in a specific way to certain of the events. It is apparent that the Examiner is equating the message sequence generator of Swift to Appellants' claimed protocol tester and the target object to the claimed item under test.

The Examiner then states that Swift teaches selecting a protocol layer to be emulated by the protocol tester for testing a specified protocol layer of the item under test on the basis of the communication procedure, citing the page 7, paragraph 1, line 1 - paragraph 2, line 9; page 6, paragraph 3; and page 8 paragraph 5 – the use of the TCP or IP or other protocol capable of transferring messages. However Swift is not emulating any of these protocols as recited by Appellants, but is merely using these protocols as a transmission medium for getting the message sequences from the message sequence generator to the target object. There is no indication in Swift that what is being tested is "a specified protocol layer of the item under test" as recited by Appellants. Swift does not emulate a protocol layer, and therefore does not "select a

protocol layer" as recited by Appellants. Swift merely indicates that in the particular embodiment using the TCP/IP communication network the message sequence engine produces TCP/IP capable applications, i.e., message sequences that are transmittable over the TCP/IP network.

The Examiner then states that Swift teaches selecting abstract communication interfaces of the emulated protocol layer for the communication procedure, citing page 7, paragraph 2, lines 1-9 – software applications that build interfaces. However Appellants find no reference in the cited portion of Swift to building interfaces, especially "abstract communication interfaces of the emulated protocol layer" as recited by Appellants. The only reference in the cited paragraph is to how the message sequence engine is implemented as a software application using a fourth generation language for developing windows graphically.

The Examiner further states that Swift teaches selecting communication data contained in description files to be exchanged at the abstract communication interfaces, citing page 9, paragraph 2. The cited paragraph indicates how a Swift user builds a message sequence definition by inputting a sequence name and having the message sequence engine load the previously saved sequence definition from the message database to the message sequence definition. It appears that the Examiner is equating the claimed description files to the message database of Swift and the claimed communication data to the saved sequence definition. However these are message sequences, not communication data that is "exchanged at the abstract communication interfaces" as recited by Appellants.

Finally the Examiner states that Swift teaches automatically setting up through the protocol tester the communication procedure on the basis of the selections made in the

above selecting steps, with parameters for the abstract communication interfaces and the communication data selecting steps being made graphically, citing page 7, paragraph 3, lines 1-5 and Fig. 4A, items 406-422 – message created, interfaces produced with PowerBuilder/PowerSockets, specific description file in Fig. 3 (message sequence definition 222). Appellants do not find any reference in the cited portion of Swift to automatic setting up through the message sequence generator of the communication procedure on the basis of the selections made, as the message sequence generator merely builds the message sequence by the user interacting graphically with the message sequence engine. The message sequence engine allows the user to create the message sequence definition and then upon the user's request to transmit the message sequence corresponding to the message sequence definition onto the network. Appellants can only assume, absent a clear statement by the Examiner, that the Examiner is equating the transmission of the message sequence corresponding to the message sequence definition to the claimed automatic setting up of the communication procedure (see Annex A). Appellants submit that there is insufficient information in Swift to arrive at such a conclusion.

Therefore claim 1 is deemed not to be anticipated by Swift since Swift neither teaches or suggests to one of ordinary skill in the art the steps of selecting a protocol layer to be emulated, selecting the abstract communication interfaces for the emulated protocol layer, selecting the communication data for exchange across the abstract communication interfaces nor the automatic setting up of the communication procedure based upon selections made. Swift is deemed merely to generate message sequences for target objects without selecting any particular protocol layer to be emulated, and uses an established communication procedure rather than setting up the

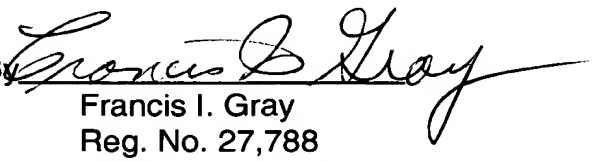
communication procedure based upon the protocol layer selections. Nowhere in Swift is there any reference to any terminology that indicates testing of a selected protocol layer, i.e., there is no reference to protocol layers or the OSI model, there is no reference to abstract communication interfaces, there is no reference to service access points, there is no reference to protocol data units and there is no reference to abstract service primitives. It appears that the Examiner is merely assuming, or taking Official Notice, of these items although he cites no appropriate reference or indicates how they actually fit within the message sequence generator of Swift.

Conclusion

Independent claims 1 and 8 are deemed to be allowable as being neither anticipated nor rendered obvious to one of ordinary skill in the art by Swift, as indicated above, and claims 2-7 and 9-13 dependent therefrom also are deemed to be allowable as depending from allowable claims. Therefore Appellants request that the Examiner's rejection of claims 1-13 be reversed, and that this case be passed to issue.

Respectfully submitted,

JOERG EHRHARDT et al

By 

Francis I. Gray
Reg. No. 27,788
Attorney for Applicant

TEKTRONIX, INC.
P. O. Box 500, MS 50-LAW
Beaverton, Oregon 97077
(503) 627-7261

7057 US

Claims Appendix

1. A method of setting up a communication procedure between instances comprising the steps of:

selecting the instances that take part in the communication procedure, one instance being a protocol tester and another instance being an item under test;

selecting a protocol layer to be emulated by the protocol tester for testing a specified protocol layer of the item under test on the basis of the communication procedure;

selecting abstract communication interfaces of the emulated protocol layer for the communication procedure;

selecting communication data contained in description files to be exchanged at the abstract communication interfaces; and

automatically setting up through the protocol tester the communication procedure on the basis of the selections made in the above selecting steps, with parameters for the abstract communication interfaces and the communication data selecting steps being made graphically.

2. The method as recited in claim 1 wherein the instances selecting step comprises the step of selecting the instances graphically, and/or the emulated protocol layer selecting step comprises the step of selecting the emulated protocol layer graphically, and the parameters selectable in these steps being assigned description files that are used in the setting up step.

3. The method as recited in claims 1 or 2 wherein the abstract communication interfaces comprise Service Access Points (SAPs).

4. The method as recited in claim 3 wherein the communication data comprise at least one type selected from the group consisting of Protocol Data Units (PDUs) and Abstract Service Primitives (ASPs).

5. The method as recited in claims 1 or 2 wherein the communication data comprise at least one type selected from the group consisting of Protocol Data Units (PDUs) and Abstract Service Primitives (ASPs).

6. The method as recited in claim 1 wherein the communication data selecting step comprises the steps of:

graphically selecting a data format; and

graphically setting up a communication sequence between the selected instances.

7. The method as recited in claim 6 wherein the graphically setting up step comprises the step of entering source code.

8. A protocol tester comprising:

means for selecting instances taking part in a communication procedure, one of the instances being the protocol tester and another instance being an item under test;

means for selecting a protocol layer to be emulated by the protocol tester for

testing a specified protocol layer of the item under test on the basis of the communication procedure;

means for selecting abstract communication interfaces of the emulated protocol layer for the communication procedure;

means for selecting communication data contained in description files to be exchanged at the abstract communication interfaces; and

means for automatically setting up the communication procedure through the protocol tester on the basis of the selections of the various selecting means, with parameters for the abstract communication interfaces and the communication data selecting means being made graphically.

9. The protocol tester as recited in claim 8 wherein the instances selecting means and/or the emulated protocol layer selecting means comprise graphical selecting means and the parameters selected by these selecting means are assigned description files that are used in the automatically setting up means.

10. The protocol tester as recited in claims 8 or 9 wherein the abstract communication interfaces comprise Service Access Points (SAPs).

11. The protocol tester as recited in claim 10 wherein the communication data comprises one type selected from the group consisting of Protocol Data Units (PDUs) and Abstract Service Primitives (ASPs).

12. The protocol tester as recited in claim 11 further comprising means for entering source codes.

13. The protocol tester as recited in claim 8 wherein all parameters selected by all the selecting means are assigned description files that are used by the setting up means.